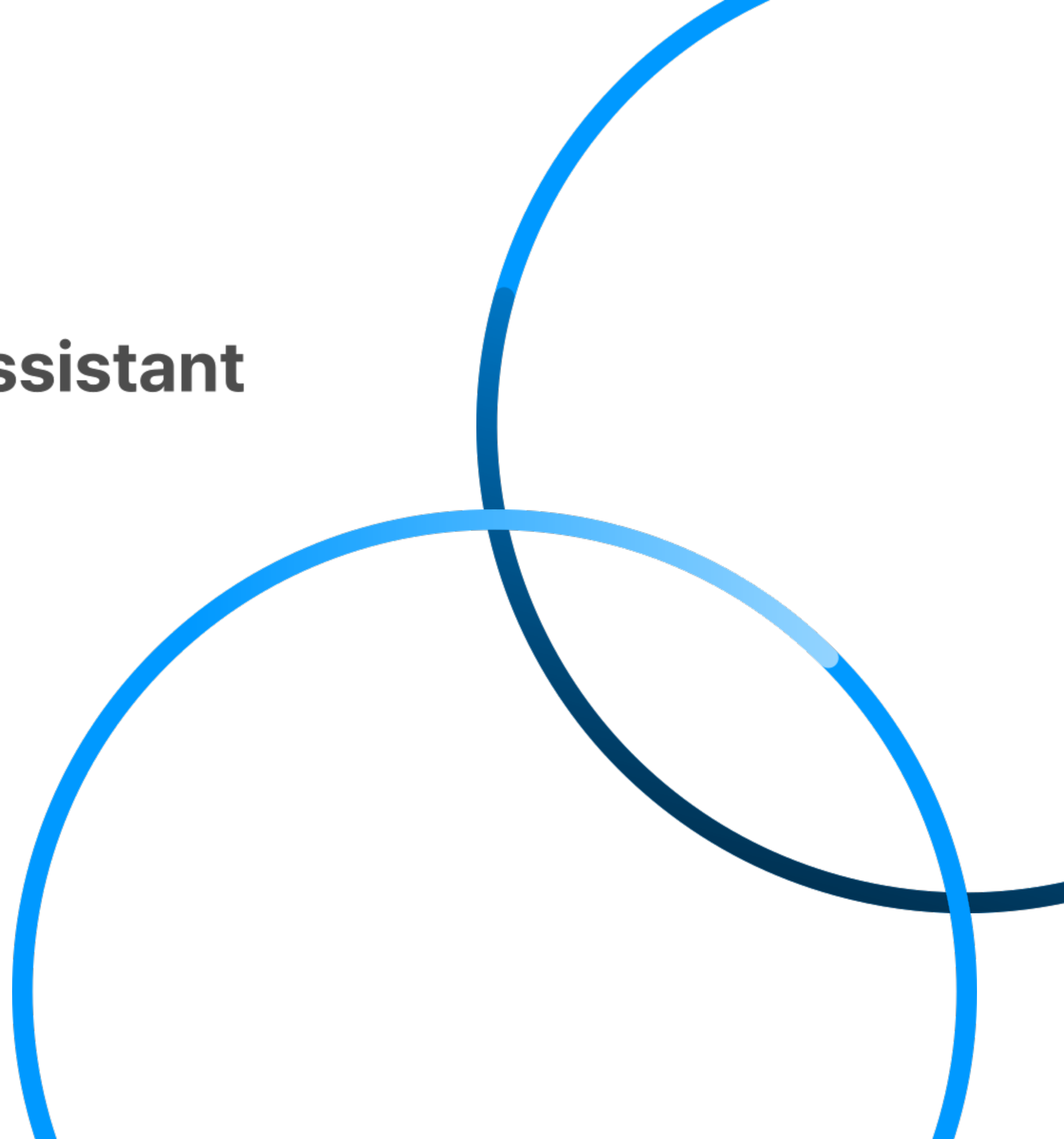


Capstone Design Project

"Mail AI" : AI-based Mail Assistant

The 3rd Bi-weekly Meeting

Team E: Yosep Kim, Soomin Lim, Dohyun Bu, Miseo Jeong



Agenda

Chapter

01

Introduction

- Timeline
- Teamwork

Chapter

02

Midterm Review

- Email Generation
- User Authentications
 - Login
 - Signup

Chapter

03

New Features

- Keyword Extration
 - each email
 - all email
- Email Summary
- CI/CD

Chapter

04

Conclusion

- Further Plans
- Q & A

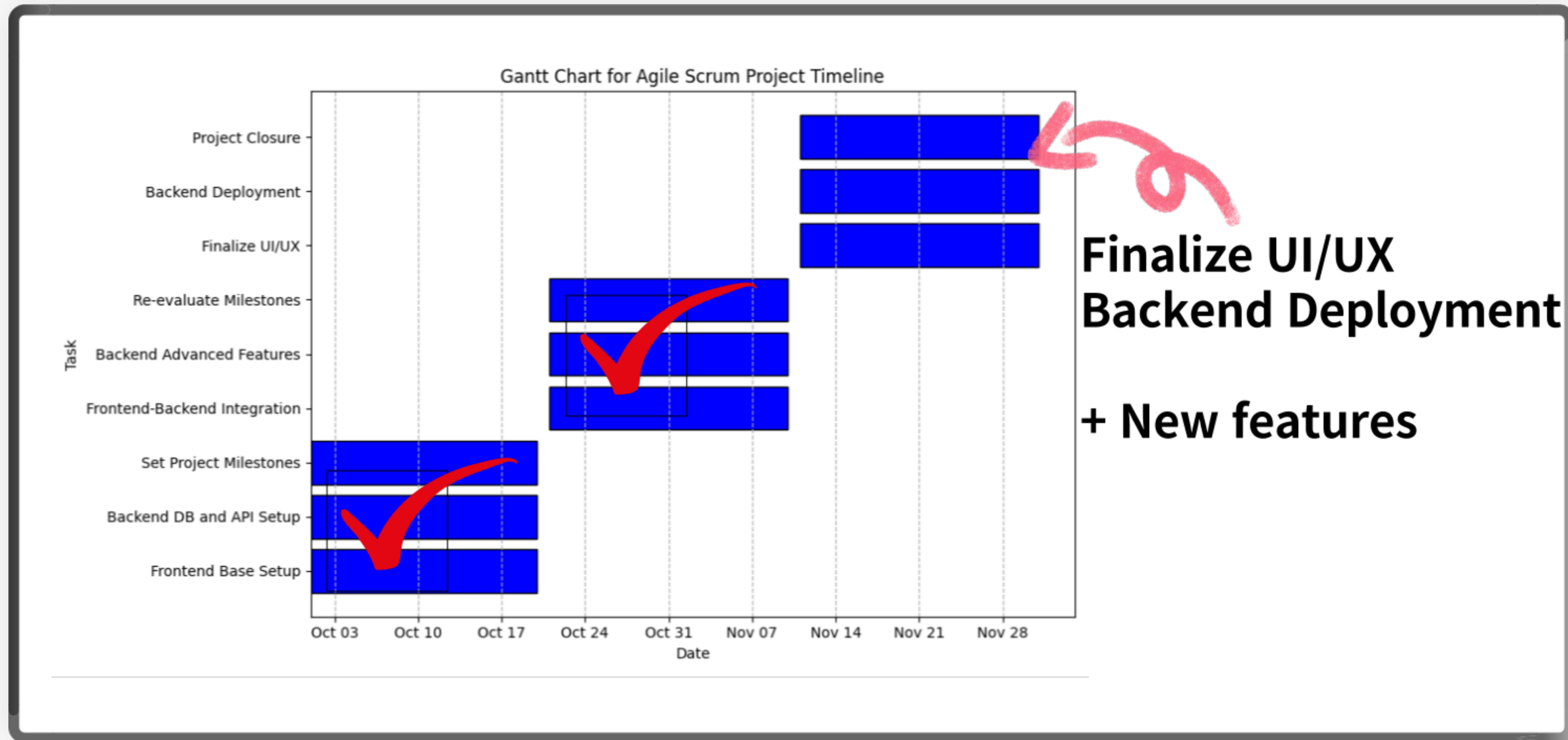
Chapter

01

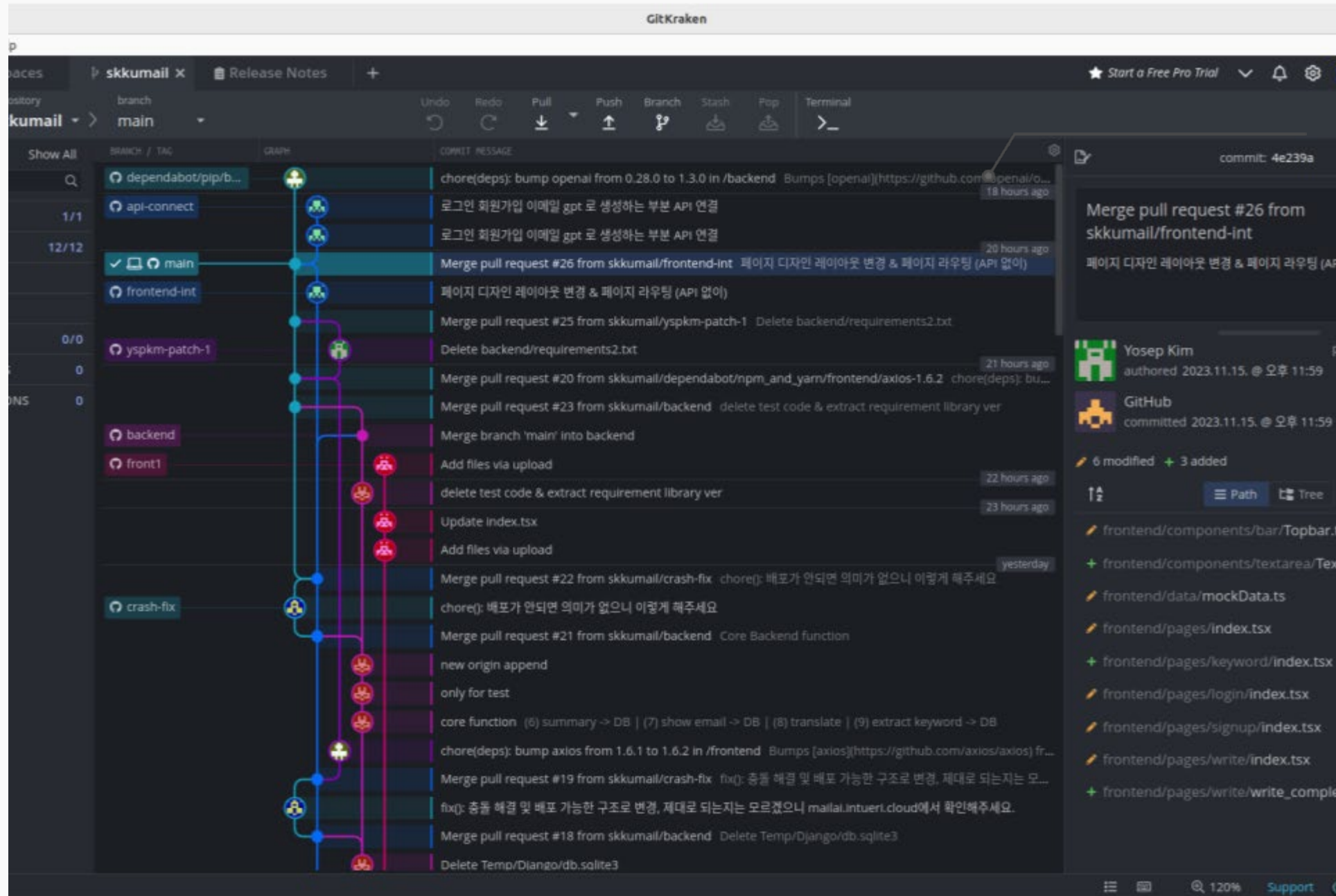
Introdction

- Timeline
- Team Role and Cooperations

Timeline: We are in 3rd Scrum!



Teamwork: Cooperation with GitHub and Git!



Team Roles

Team Member	Role & Responsibilities
Yosep Kim	Project Manager, Backend Development
Sumin Lim	Frontend Development (Next.js, UX/UI)
Dohyeon Boo	Backend Development (FastAPI, GPT, DB)
Miseo Jeong	Frontend Development (Next.js, UX/UI)

Commits

Branches

Merges

93

12

32

Chapter

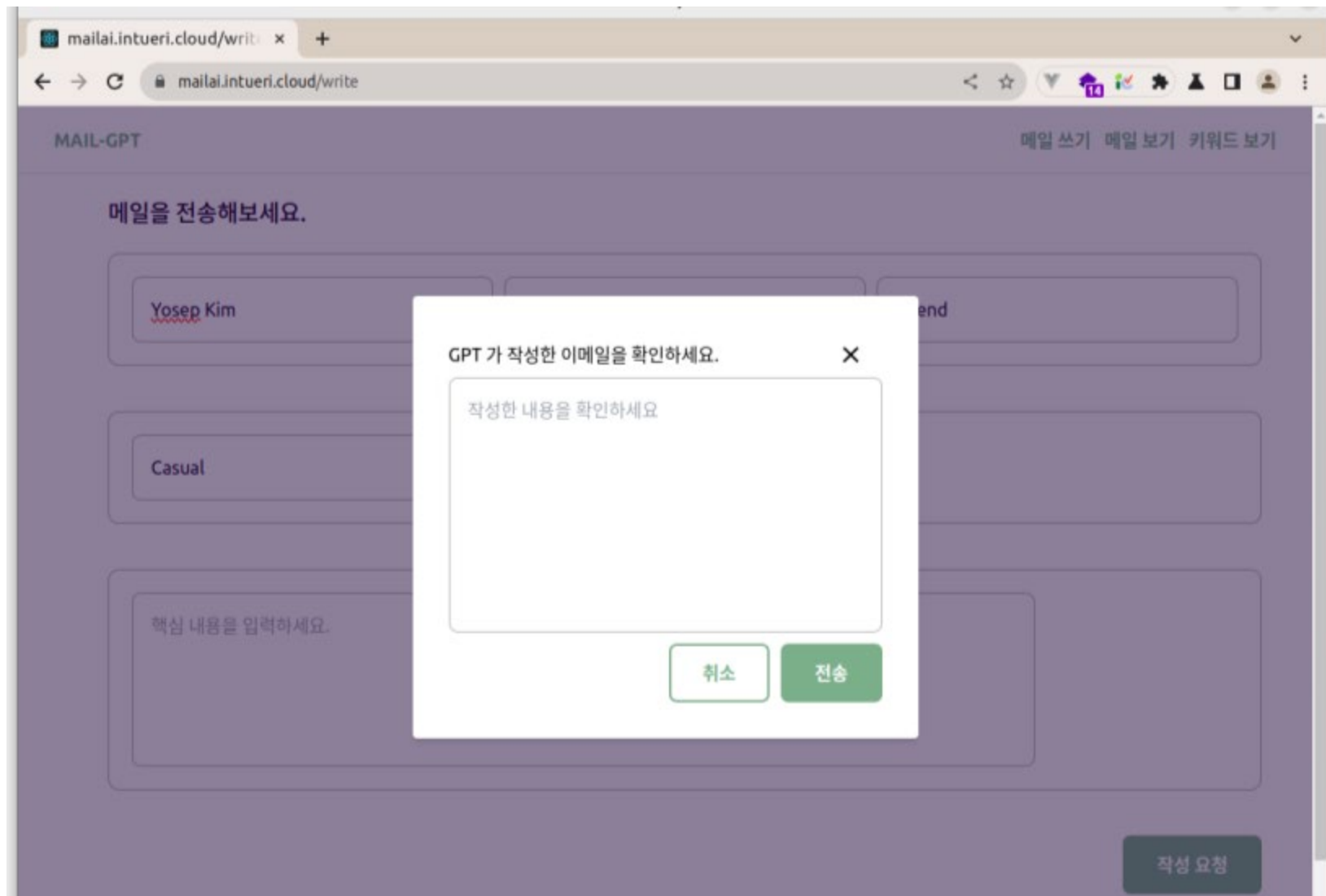
02

Midterm Review

- Email Generation
- User Authentications
 - Login
 - Signup



Email Generation: Frontend & Demo



MAIL-GPT

메일을 전송해보세요.

수신인을 입력하세요.

수신인 이메일을 입력하세요.

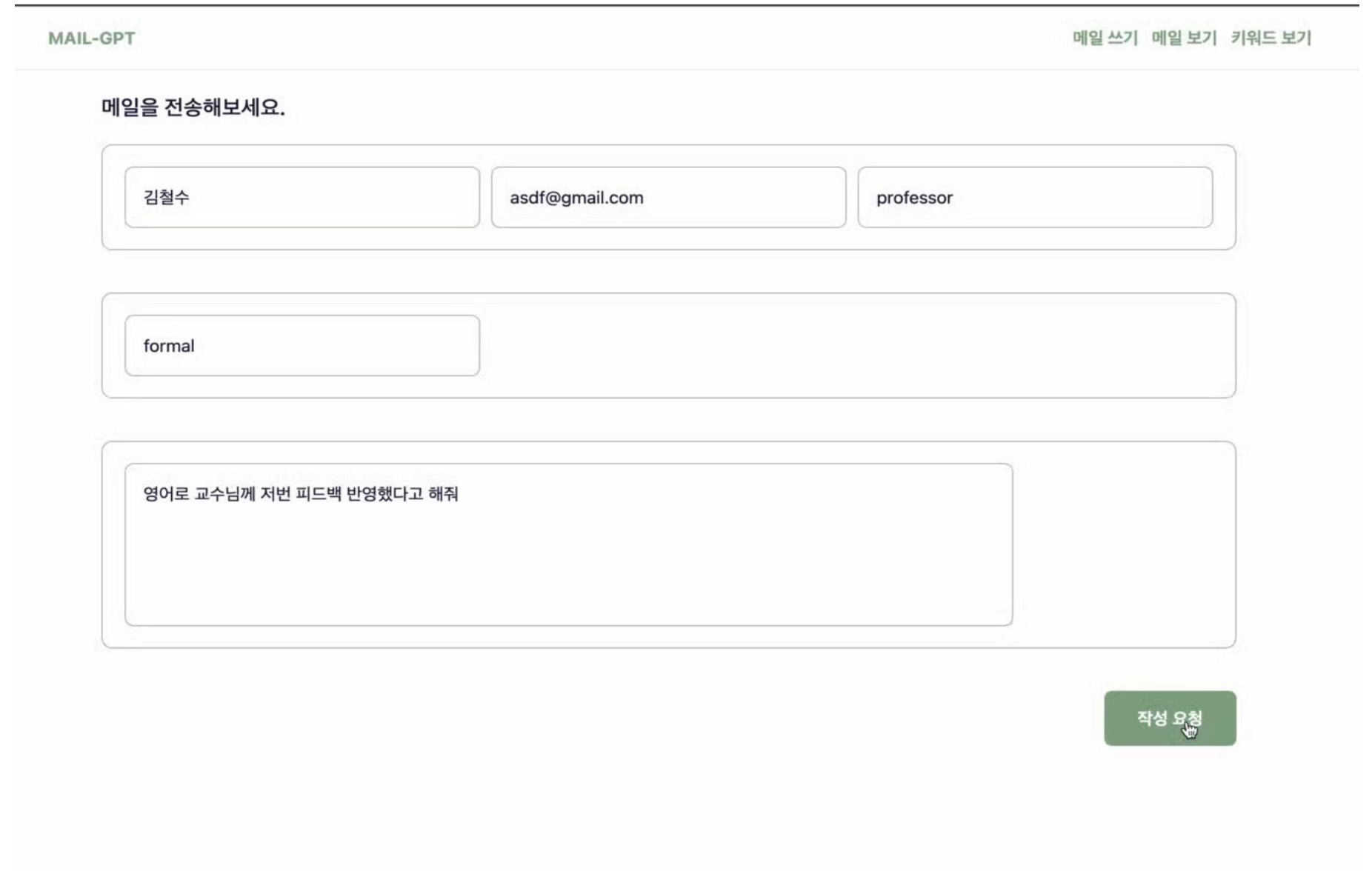
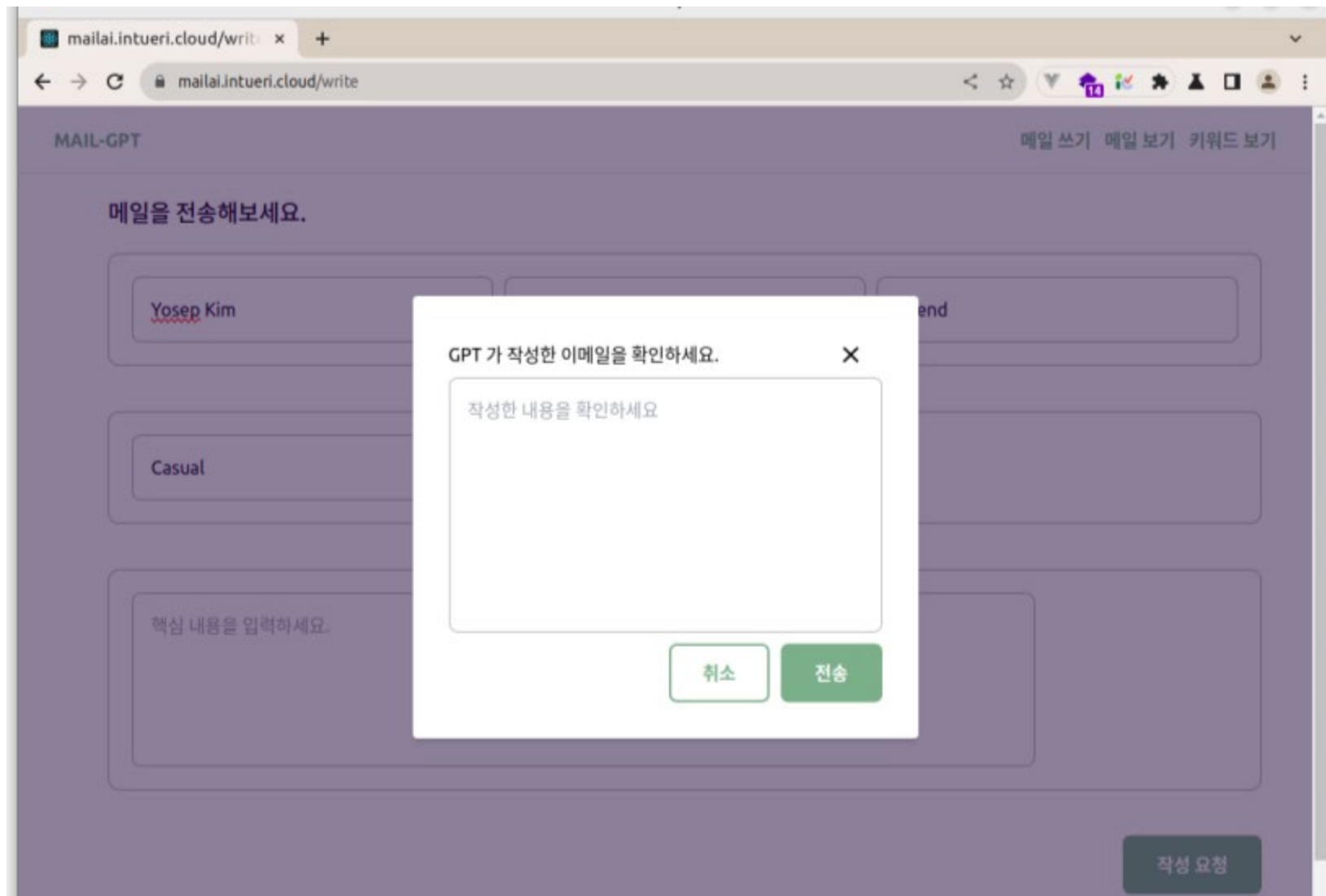
수신인과의 관계를 입력하세요.

메일 작성 스타일을 입력하세요.

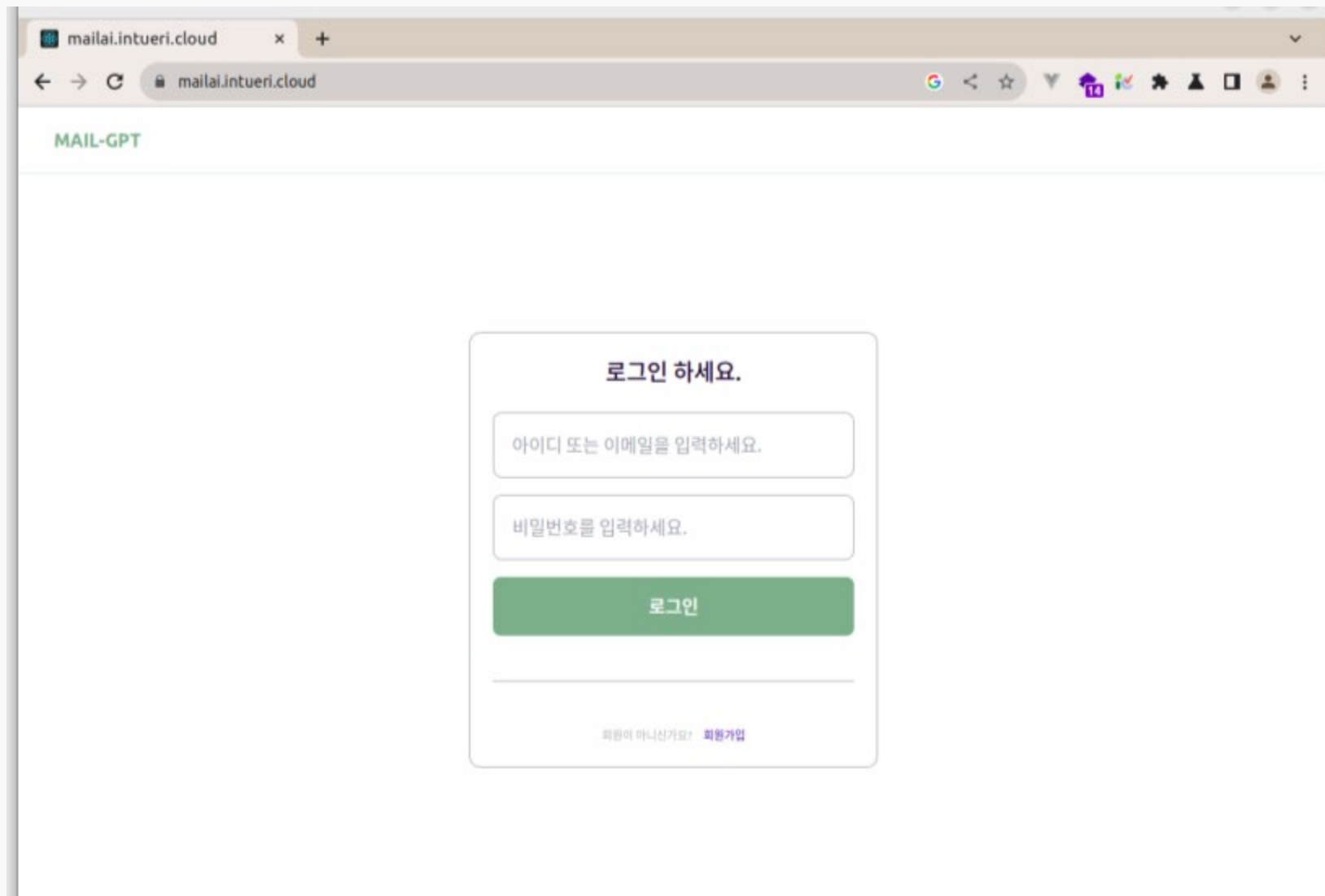
핵심 내용을 입력하세요.

작성 요청

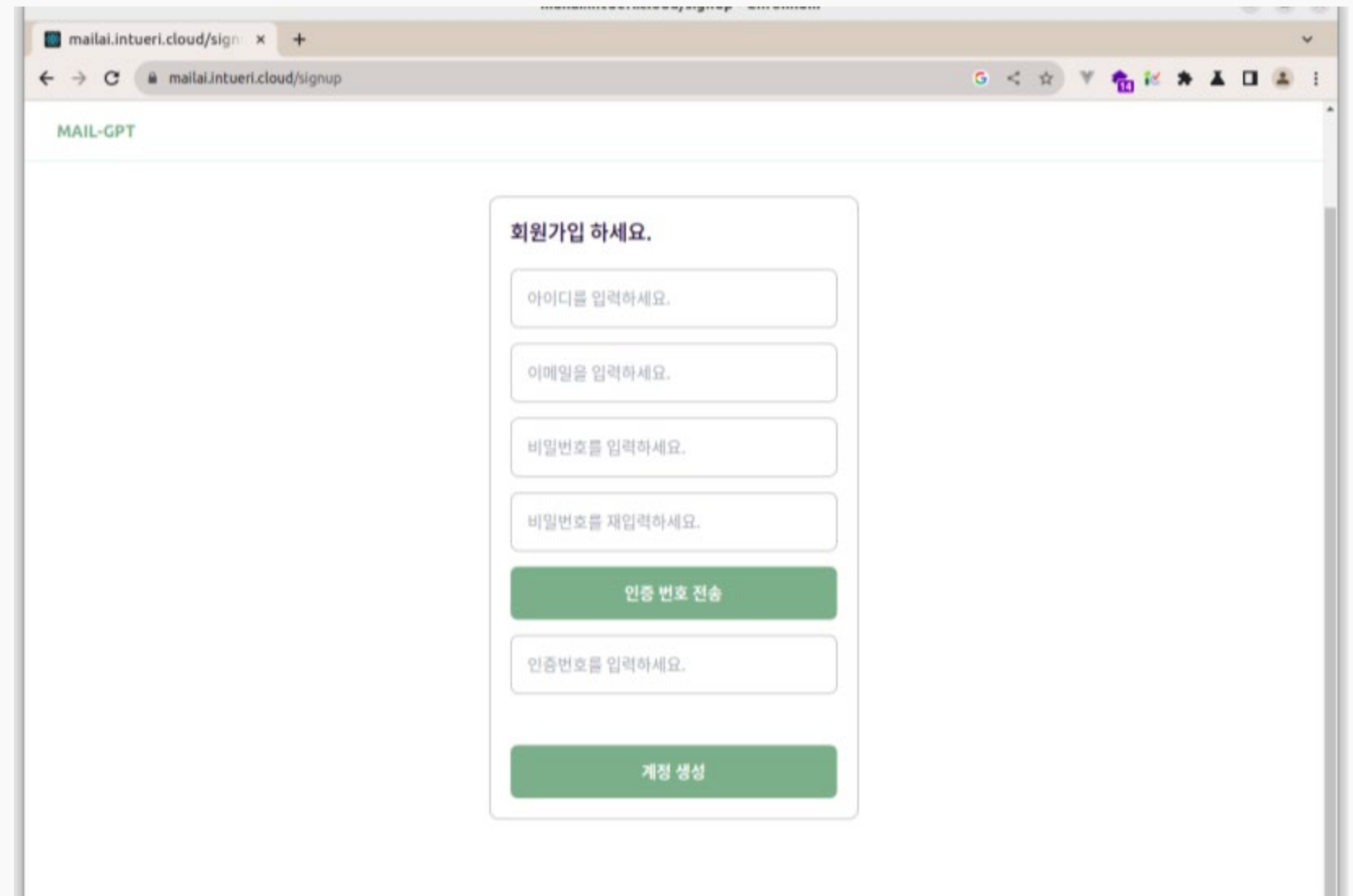
Email Generation: Frontend & Demo



User Authenticaiton: Login & Signup



The screenshot shows the login page of the mailai.intueri.cloud application. The browser's address bar displays 'mailai.intueri.cloud'. The page header includes the text 'MAIL-GPT'. The main content area features a login form with the title '로그인 하세요.' (Please log in). The form contains two input fields: '아이디 또는 이메일을 입력하세요.' (Enter ID or email) and '비밀번호를 입력하세요.' (Enter password). Below these fields is a green button labeled '로그인' (Login). At the bottom of the form, there is a link that reads '회원이 아니신가요? 회원가입' (Are you not a member? Sign up).



The screenshot shows the signup page of the mailai.intueri.cloud application. The browser's address bar displays 'mailai.intueri.cloud/signup'. The page header includes the text 'MAIL-GPT'. The main content area features a signup form with the title '회원가입 하세요.' (Please sign up). The form contains four input fields: '아이디를 입력하세요.' (Enter ID), '이메일을 입력하세요.' (Enter email), '비밀번호를 입력하세요.' (Enter password), and '비밀번호를 재입력하세요.' (Re-enter password). Below these fields is a green button labeled '인증 번호 전송' (Send verification code). Further down is another input field labeled '인증번호를 입력하세요.' (Enter verification code), followed by a green button labeled '계정 생성' (Create account).

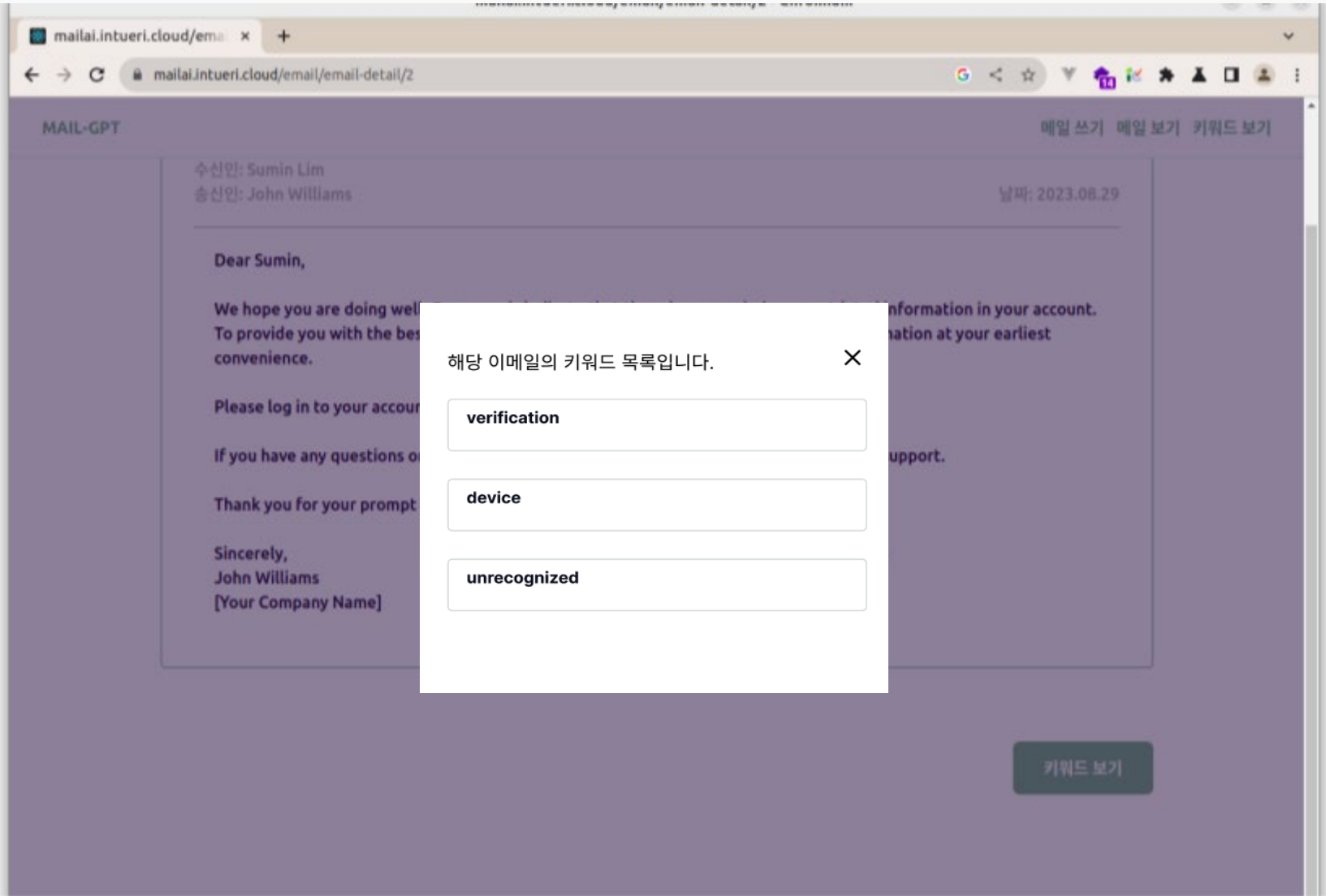
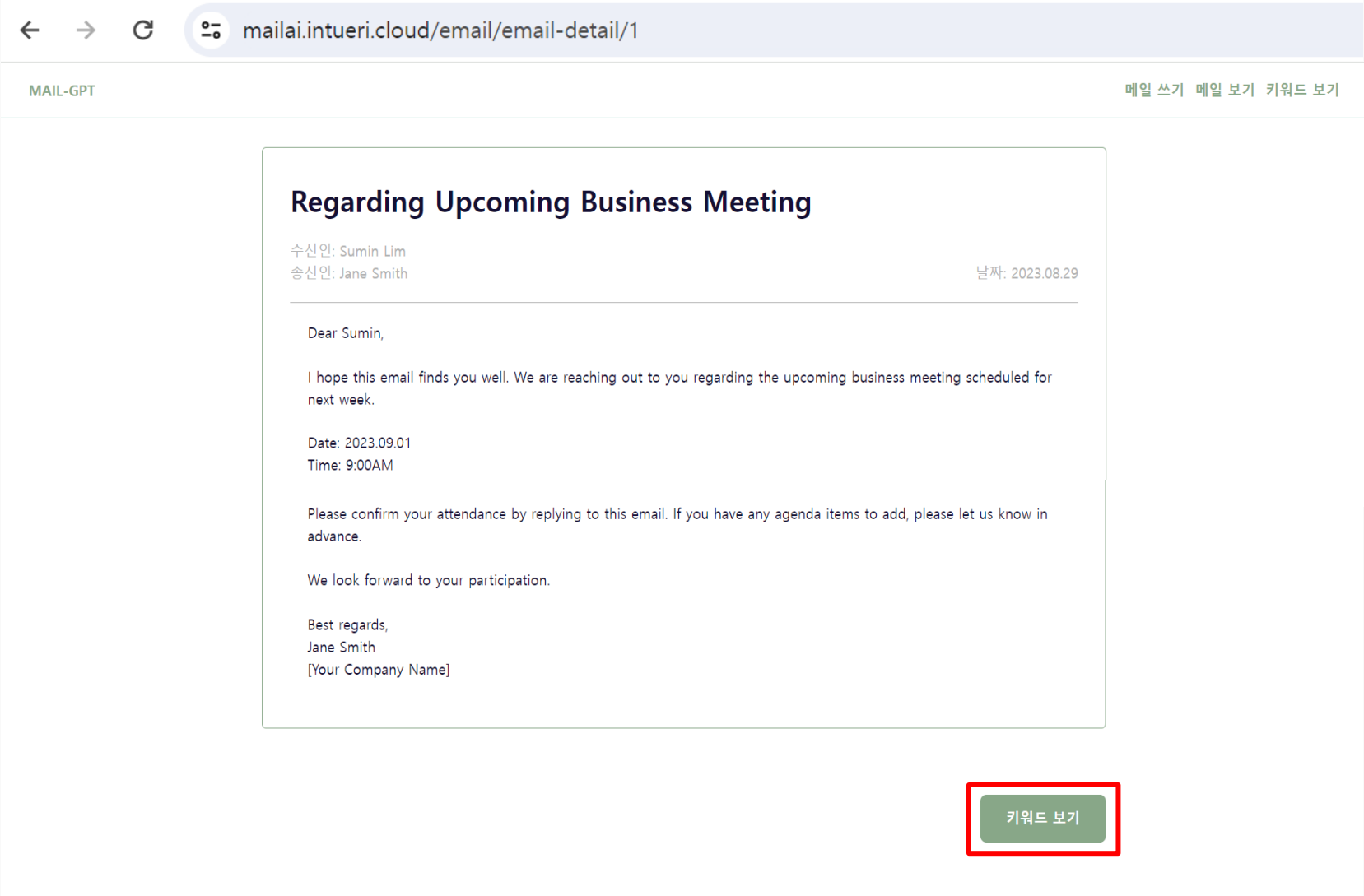
Chapter 03

New Features

- Keyword Extration
 - each email
 - all email
- Email Translation
- CI/CD

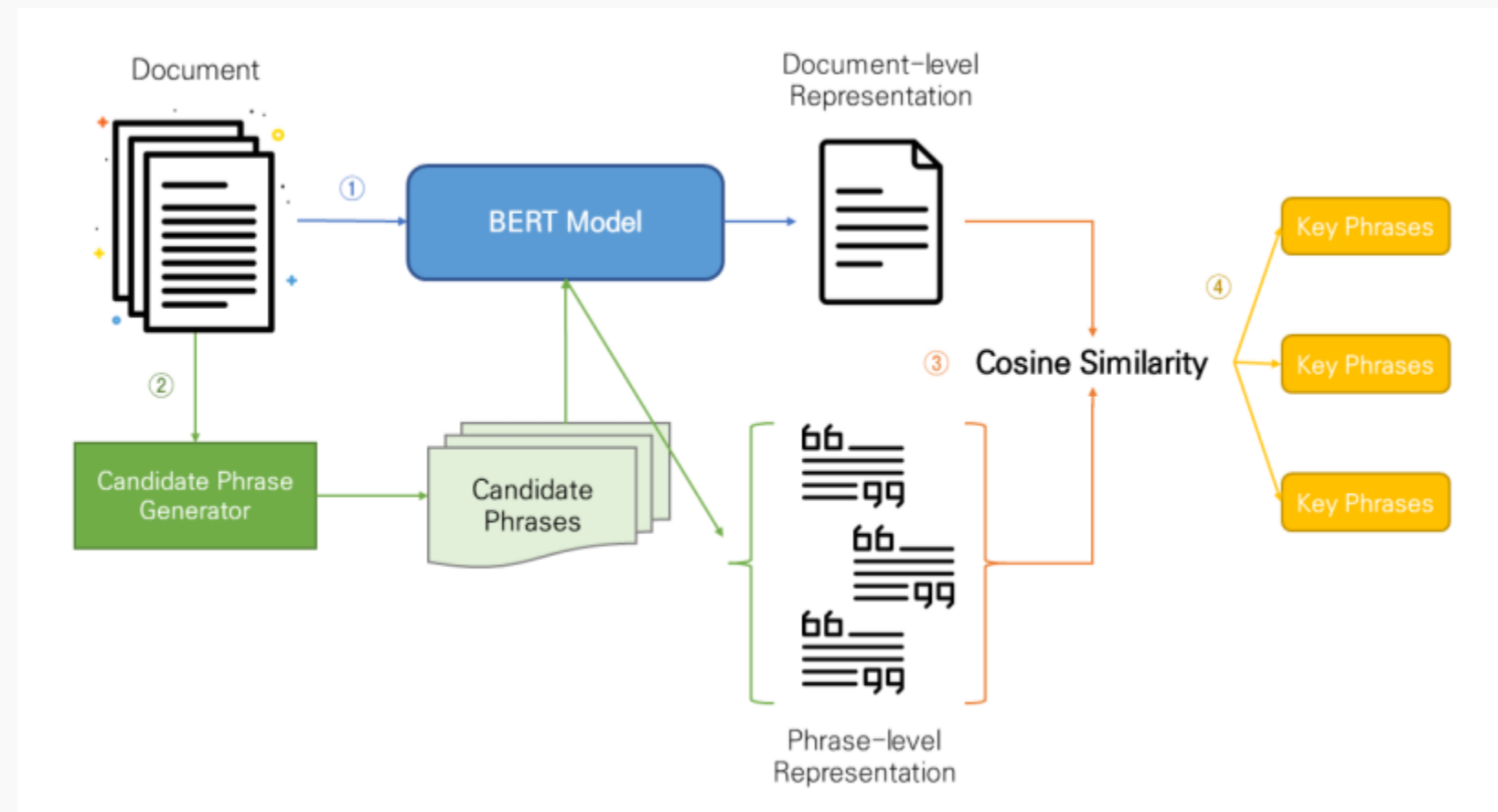


Keyword Extraction: Each & All



Keyword Extraction: Architecture

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a 'reasonable' way (see inductive bias).



```

    )
    # KeyBERT를 사용하여 키워드 추출
    keywords = keybert_model.extract_keywords(
        email_text, keyphrase_ngram_range=(1, 1), top_n = 5, stop_words="eng
    )

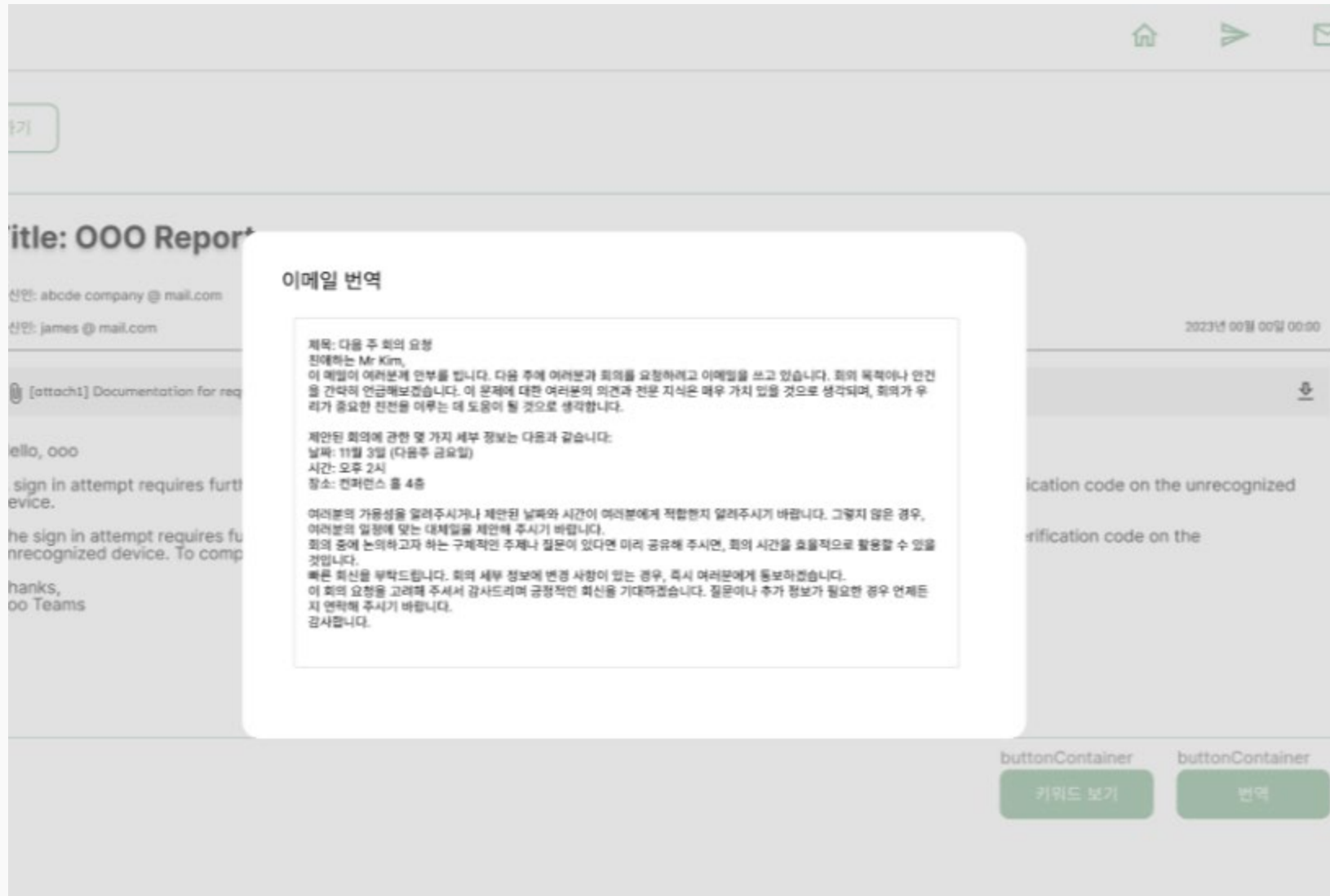
    total_list = []
    for i in range(len(keywords)):
        total_list.append(keywords[i][0])

    temp_user_msg.keyword = str(total_list)
    db.add(temp_user_msg)
    db.commit()

    # 추출된 키워드를 JSON 형식으로 반환
    return JsonResponse({'message': '연결 성공', "data":total_list})
except Exception as e:
    # 오류가 발생한 경우 500 Internal Server Error 반환
    raise HTTPException(status_code=500, detail=f"An error occurred: {str(e)}")

```

Email Translation: Google Translator API



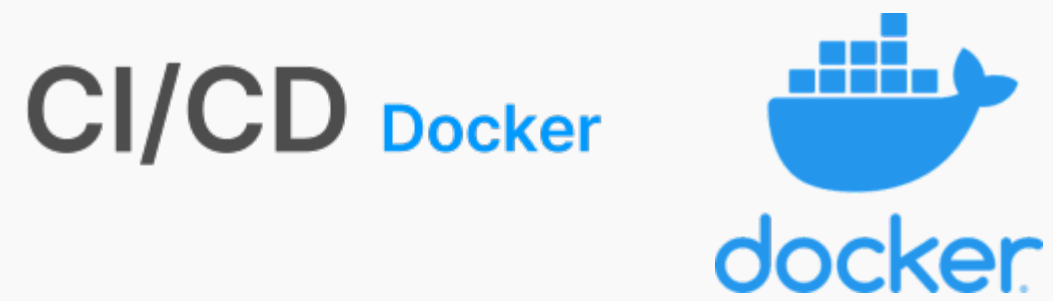
```
##### Translate #####
@app.post("/translate")
async def translate_text(request: Request, db: Session=Depends(get_db)):
    try:
        data = await request.json()

        pre_text = data.get('contents')

        translator = Translator()
        result = translator.translate(pre_text, dest='ko').text

        return JSONResponse({'message': '연결 성공', 'data':result})

    except Exception as e:
        return JSONResponse({'message': '연결 fail', 'data':str(e)})
```

```
FROM rockylinux/rockylinux:9.2

RUN dnf -y update && \
    dnf -y install gcc make python3 python3-pip python3-devel \
    dnf clean all

WORKDIR /app

COPY requirements.txt .

RUN pip3 install --no-cache-dir -r requirements.txt

COPY . .

CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8000"]
```

```
FROM ubuntu:jammy-20231004

RUN apt-get update \
    && apt-get upgrade -y \
    && apt-get install -y ca-certificates curl gnupg \
    && mkdir -p /etc/apt/keyrings \
    && curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key \
    | gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg

ENV NODE_MAJOR=20

RUN echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesource.com/node_${NODE_MAJOR}.x nodistro" \
    | tee /etc/apt/sources.list.d/nodesource.list \
    && apt-get update \
    && apt-get install nodejs -y

WORKDIR /app
COPY package.json .

RUN npm install

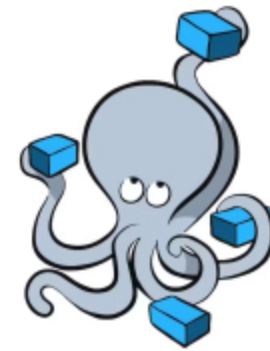
COPY . .
RUN npm run build

EXPOSE 3000

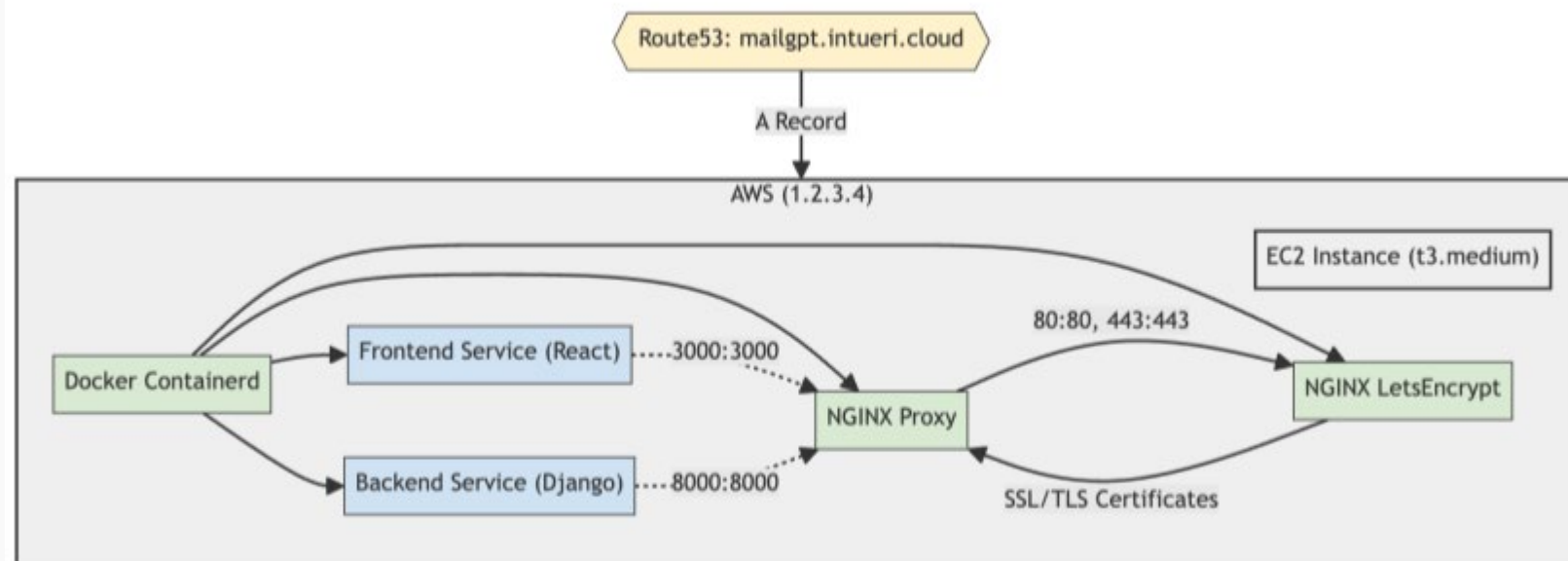
CMD ["npm", "start"]
```



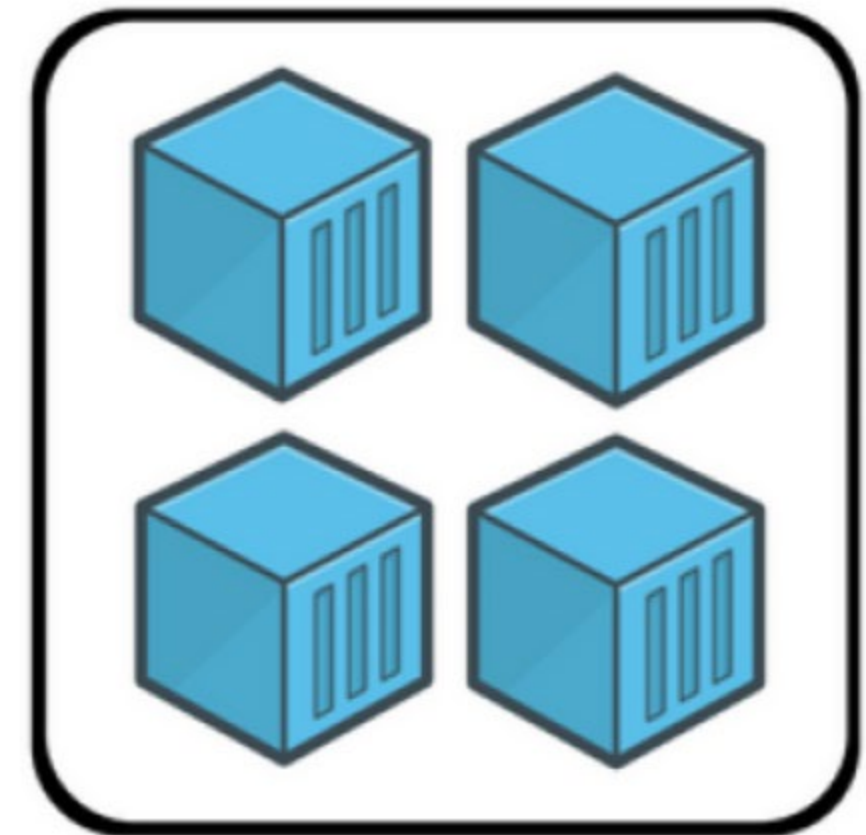
CI/CD Docker Compose



docker
Compose



Docker



Docker Compose

Chapter 04

Conclusion

- Further Plans
- Q & A

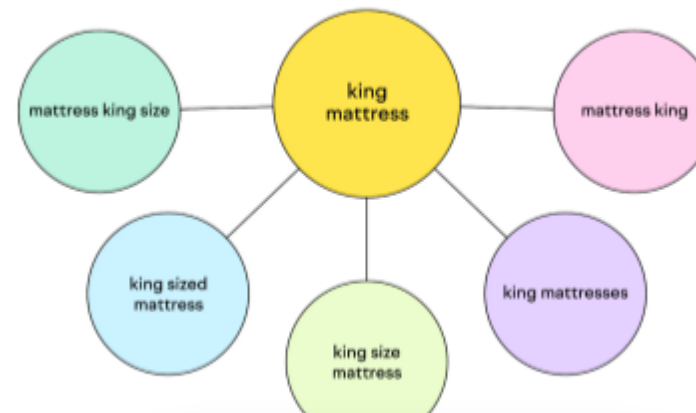


Further Plan ●

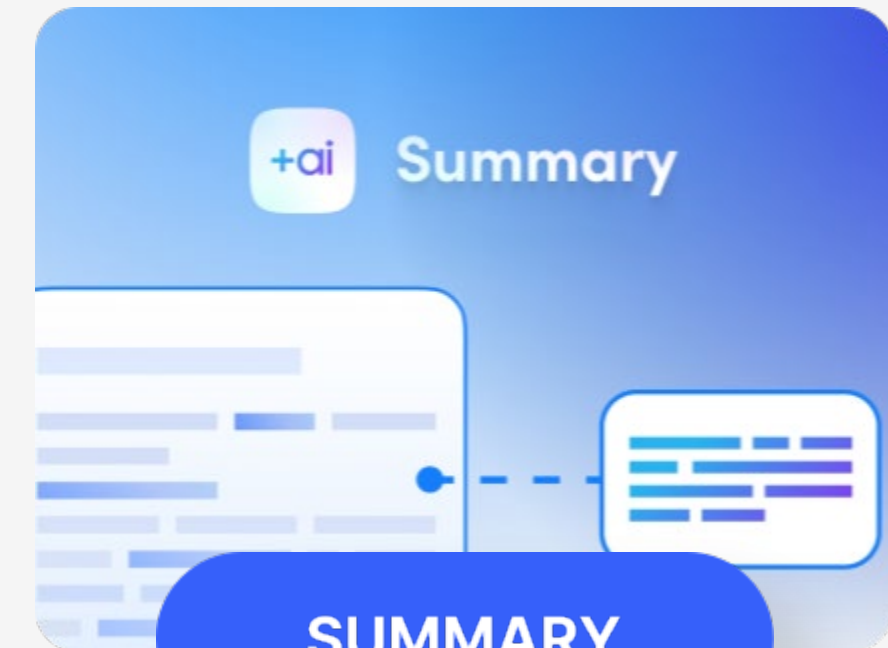


EMAIL DB

Keyword Cluster Example



CLUSTERING



SUMMARY



Capstone Design Project

**THANK YOU
FOR WATCHING!**

